

(19) 世界知的所有権機関
国際事務局(43) 国際公開日
2002 年 9 月 6 日 (06.09.2002)

PCT

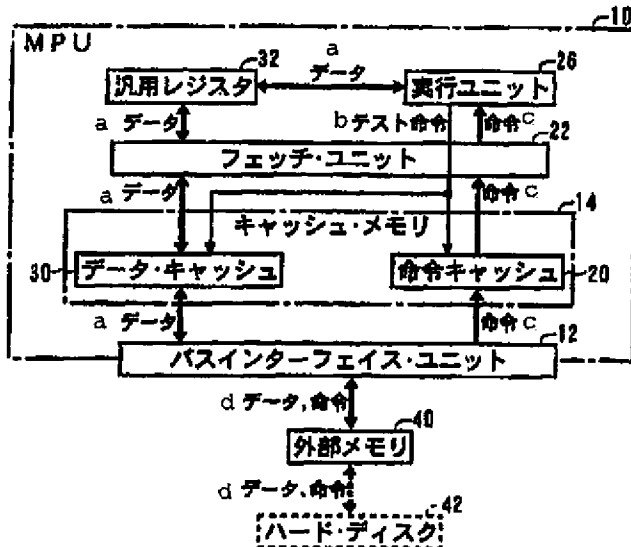
(10) 国際公開番号
WO 02/069150 A1

- (51) 国際特許分類⁷: G06F 12/08, 9/30, 9/38, 9/32, 9/34 CORPORATION [US/US]; 10504 ニューヨーク州
アーモンク NY (US).
- (21) 国際出願番号: PCT/JP02/01272
- (22) 国際出願日: 2002 年 2 月 14 日 (14.02.2002)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願2001-51539 2001 年 2 月 27 日 (27.02.2001) JP
- (71) 出願人 (米国を除く全ての指定国について): インター
ナショナル・ビジネス・マシーンズ・コーポレー
ション (INTERNATIONAL BUSINESS MACHINES
- (72) 発明者: および
- (75) 発明者/出願人 (米国についてのみ): 上田 真
(UEDA, Makoto) [JP/JP]; 〒520-2362 滋賀県 野洲郡野
洲町 大字市三宅800番地 日本アイ・ビー・エム株式会社 野
洲事業所 Shiga (JP).
- (74) 代理人: 坂口 博, 外 (SAKAGUCHI Hiroshi et al.); 〒
242-8502 神奈川県 大和市 下鶴間1623番地14 日本ア
イ・ビー・エム株式会社 大和事業所 Kanagawa (JP).
- (81) 指定国 (国内): AE, AG, AL, AM, AT, AU, AZ, BA, BB,
BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK,
DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU,
ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS,

[続業有]

(54) Title: MICROPROCESSOR AND INSTRUCTION EXECUTION ORDER SCHEDULING METHOD

(54) 発明の名称: マイクロプロセッサ及び命令実行順序スケジューリング方法



(57) Abstract: The degradation of the ratio of utilization of an MPU caused by mishit of a cache memory is suppressed. A microprocessor (10) has an execution unit (26) for executing instructions including an additional test instruction to check if a desired routine including instructions or a desired data structure containing data is present in cache memories (20, 30). Right before reading such a routine, a routine included in such a data structure and parallel processable, or such a data structure, the microprocessor (10) confirms whether the routine or data structure to be read is present in the cache memories (20, 30). The routine or data structure in the cache memories (20, 30), out of the routines parallel processable and data structures, is preferentially executed by the microprocessor (10).

- 32...GENERAL PURPOSE REGISTER 30...DATA CACHE
a...DATA 20...INSTRUCTION CACHE
26...EXECUTION UNIT 12...BUS INTERFACE UNIT
b...TEST INSTRUCTION 40...EXTERNAL MEMORY
c...INSTRUCTION 42...HARD DISC
22...FETCHING UNIT d...DATA INSTRUCTION
14...CACHE MEMORY

[続業有]



LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

LU, MC, NL, PT, SE, TR), OAPI 特許 (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告書

(84) 指定国 (広域): ARIPO 特許 (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア特許 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT,

2 文字コード及び他の略語については、定期発行される各 PCT ガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

(57) 要約:

キャッシュ・メモリのヒット・ミスによる MPU の利用率の低下を低減させる。 実行ユニット 26 が実行する命令に、複数の命令を含む所要のルーチン又は複数のデータを含む所要のデータ構造体がキャッシュ・メモリ 20, 30 内に存在するか否かを確認するテスト命令が追加されたマイクロプロセッサ 10 を用いて、前記ルーチン又はデータ構造体に含まれる並列処理が可能なルーチン又はデータ構造体を読み出す直前に、その読み出そうとしているルーチン又はデータ構造体がキャッシュ・メモリ 20, 30 内に存在するか否かをマイクロプロセッサ 10 に確認させ、並列処理が可能なルーチン又はデータ構造体の中で、キャッシュ・メモリ 20, 30 に存在するルーチン又はデータ構造体から優先的にマイクロプロセッサ 10 に処理させる。

明細書

マイクロプロセッサ及び命令実行順序スケジューリング方法

5 技術分野

本発明は、マイクロプロセッサ及び命令実行順序スケジューリング方法に関し、より詳しくは、プログラムが指示する順序で命令を実行するマイクロプロセッサ及びマイクロプロセッサの命令実行順序のスケジューリング方法に関する。

10

背景技術

図4にMPU(マイクロプロセッサ・ユニット)10'の一構成例を示す。MPU10'は、外部メモリ40よりも小容量かつ高速アクセス可能で、外部メモリ40から読み出される命令の一部及びデータの一部が記憶されるキャッシュ・メモリ14と、キャッシュ・メモリ14又は外部メモリ40から命令又はデータを読み出すフェッチ・ユニット22と、読み出された命令を実行する実行ユニット26'と、実行中の命令が使用するデータが記憶される汎用レジスタ32と、外部デバイス(40)が接続されるバスインターフェイス・ユニット12を含む。

20 キャッシュ・メモリ14は、命令が記憶される命令キャッシュ20とデータが記憶されるデータ・キャッシュ30とを含む。MPU10'はバスインターフェイス・ユニット12を介して外部メモリ(半導体記憶装置)40に接続されており、外部メモリ40とMPU10'間で命令及びデータの読み書きが行われる。ただし、キャッシュ・メモリ20,30の
25 アクセス速度は外部メモリ40よりも60~100倍高速なので、外部メモリ40よりもキャッシュ・メモリ20,30の方を優先的に利用する。

MPU10'が必要とする命令又はデータがキャッシュ・メモリ20, 30に存在しない場合は、外部メモリ40から命令又はデータを読み出す。このキャッシュ・メモリ20, 30に存在しない場合の外部メモリ40からの命令又はデータの読み出しはハードウェアで制御される。例えば、MPU10'全体の制御を行うコントロール・ユニット(図示していない)がこの制御を行う。

外部メモリ40はハード・ディスク(固定磁気記憶装置)42とも接続されており、外部メモリ40とハード・ディスク42間で命令及びデータの読み書きが行われる。MPU10'が必要とする命令又はデータが外部メモリ40に存在しない場合は、ハード・ディスク42から命令又はデータを読み出す。この外部メモリ40に存在しない場合のハード・ディスク42からの命令又はデータの読み出しはソフトウェアで制御される。通常はOS(オペレーティング・システム)がこの制御を行う。

MPU10'に命令を実行させる場合、フェッチ・ユニット22がその命令を命令キャッシュ20又は外部メモリ40から読み出す。命令キャッシュ20に目的の命令が存在すれば命令キャッシュ20からその命令を読み出し、無ければ外部メモリ40から読み出す。外部メモリ40から命令を読み出した場合、読み出した命令は命令キャッシュ20にも送られ記憶される。

フェッチ・ユニット22に読み出された命令は、実行ユニット26'に送られ実行される。命令の実行に必要なデータは、データ・キャッシュ30又は外部メモリ40から汎用レジスタ32に読み出される。データ・キャッシュ30に目的のデータが存在すればデータ・キャッシュ30からそのデータを読み出し、無ければ外部メモリ40から読み出す。外部メモリ40から読み出したデータはデータ・キャッシュ30にも送

られ記憶される。図5(a)に、2つのデータ(DATAsA, DATAsB)を使用する2種類のルーチン(FuncA, FuncB)をMPU10'に実行させるプログラムを例としたフロー・チャートを示す。図5(a)では、主にデータ(DATAsA, DATAsB)の読み出し及び処理部分を図示している。MPU10'は、図5(a)に示すプログラムで指示された実行順序でデータの読み出し及び処理を行う。

DATAsAとDATAsBは、それぞれ幾つかのデータ(DATA-A0, DATA-A1, DATA-A2, ……と、DATA-B0, DATA-B1, DATA-B2, ……)から構成されたデータ構造体である。これらDATAsAとDATAsBは各々独立したデータである。

FuncA, FuncBは、プログラムの一部分を構成する、ある機能をもった一連の命令群である。FuncAとFuncBは、それぞれ幾つかの命令(Inst-A0, Inst-A1, Inst-A2, ……と、Inst-B0, Inst-B1, Inst-B2, ……)から構成されている。FuncAとFuncBは各々独立した命令であり、例えば、分岐命令が無い限りFuncAはInst-A0, Inst-A1, Inst-A2, ……の順に命令を実行し、FuncBはInst-B0, Inst-B1, Inst-B2, ……の順に命令を実行する。

図5(a)に示すように、MPU10'は、DATAsAを読み出して(S172)、DATAsAを使用したFuncAを実行し(S174)、次にDATAsBを読み出して(S176)、DATAsBを使用したFuncAを実行する(S178)。

続いて、DATAsAを読み出して(S172')、DATAsAを使用したFuncBを実行し(S182)、次にDATAsBを読み出して(S176')、DATAsBを使用したFuncBを実行する(S186)。

図5(a)に示す順序でデータを読み出す場合、読み出そうとしているデータがデータ・キャッシュ30に無ければ、MPU10'の待ち時間が増加する。例えばDATAsAを読み出してFuncA(S174)を実行する場合、もしDATAsAがデータ・キャッシュ30に無ければ、外部メモリ40

からDATAs Aを読み出す。外部メモリ 40のアクセス速度はデータ・キャッシュ 30よりも60～100倍遅いので、この外部メモリ 40からのデータ読み出しによるMPU 10'の待ち時間は60～100倍長くなる。

- 5 しかも、仮にDATAs Bがデータ・キャッシュ 30内に存在し、DATAs Bの方がDATAs Aよりも短時間(1/100～1/60の時間)で読み出せる状態であっても、DATAs Aを外部メモリ 40から読み出して(S 172)、この読み出したDATAs Aを使用したFunc A(S 174)が実行し終わるまで、DATAs Bの読み出し(S 176)及びFunc B(S 178)を実行することはできない。
- 10

DATAs AとDATAs Bとは独立したデータであり、Func AとFunc Bとは独立した命令であるので、プログラムの実行順序を図 5(a)のフロー・チャートから図 5(b)に示すフロー・チャートに変更することも可能である。MPU 10'は、DATAs Aを読み出して(S 172)Func A及びFunc Bを実行し(S 174, S 182)、次にDATAs Bを読み出して(S 176)Func A及びFunc Bを実行する(S 178, S 186)。しかし、この場合も図 5(a)と同様に、読み出そうとしているデータがデータ・キャッシュ 30に無ければ、MPU 10'の待ち時間が増加する。

- 以上、DATAs A, DATAs Bの読み出しを例にして説明したが、Func A, Func Bを実行する際のFunc A, Func Bの読み出しも同様である。図 6に、2つのルーチン(Func A, Func B)をMPU 10'に実行させるプログラムを例としたフロー・チャートを示す。MPU 10'は、Func Aを読み出して(S 190)実行し(S 192)、次にFunc Bを読み出して(S 194)実行する(S 196)。
- 20

このFunc A, Func Bの実行の場合も、上述したデータ(図 5(a), (b))の場合と同様に、命令キャッシュ 20に目的のルーチンが無ければ、M

MPU 10の待ち時間が長くなり、MPU 10の利用率は低下する。

このようなキャッシュ・メモリ 20, 30のヒット・ミスによるMPU 10'の待ち時間の増加を減少させる方法として、実行中の処理と並行して、プログラムが近い将来必要になると予想した命令又はデータをあらかじめMPU 10'に読み出しておくプリフェッチ(先読み)がある。

プリフェッチには、例えばタッチ命令を使用する。タッチ命令は、命令又はデータの読み出しをフェッチ・ユニット 22に指示する命令である。タッチ命令が実行されると、タッチ命令で要求された命令又はデータが外部メモリ 40からキャッシュ・メモリ 20, 30に読み出される。

タッチ命令の実行中であっても、実行ユニット 26'は他の命令を並列して実行することができる。タッチ命令を用いることで、プログラム側からMPU 10'に、近い将来アクセスを行うと予想される命令又はデータを知らせることができる。MPU 10'はタッチ命令で知らされた命令又はデータをキャッシュ・メモリ 20, 30に予め読み出しておくことで、キャッシュ・メモリ 20, 30のヒット率の向上を図る。

しかし、プリフェッチする命令又はデータの予測は、通常プログラムを実行する以前のソース・プログラムの状態で行われる。プログラムの実行以前に必要なと予想された命令又はデータを読み出すため、プリフェッチされた命令又はデータが本当に必要になるとは限らない。プリフェッチが有効に機能するかどうかはプログラム実行前の予測の精度に左右され、必ず効果が現れるわけではない。

プログラムの実行途中で、OSによって命令実行順序の変更を行うマルチ・スレッドと呼ばれる方法もある。実行中のスレッドが待ち状態になると、スケジューラが実行可能な他のスレッドを実行状態に切り換える。スレッドはプログラムを構成する実行順序の入れ替えが可能な単位であり、各スレッドはプログラムの実行状態に関するコンテキストと呼

ばれる情報を持つ。OSがスレッド単位で実行順序を入れ替えるときには、コンテキスト・スイッチと呼ばれるコンテキストのレジスタへの退避及び復帰が発生する。コンテキスト・スイッチの実行には、実行中のスレッドが待ち状態になったことを示す割り込み、スケジューラの起動、
5 レジスタへのアクセス、スレッドの実行切り換えが伴う。ハード・ディスクへのアクセス待ち時間にコンテキスト・スイッチを実行する場合は、コンテキスト・スイッチの実行時間は十分短く、マルチ・スレッドは有効に機能する。しかし、キャッシュ・ミスによる待ち時間にコンテキスト・スイッチを実行する場合は、コンテキスト・スイッチの実行時間は
10 短くなく、マルチ・スレッドは有効に機能しない。

プログラムの実行途中で、MPU内部で命令実行順序の変更を行うアウト・オブ・オーダーと呼ばれる方法もある。アウト・オブ・オーダーは、複数の実行ユニットを用いた並列処理を行うスーパー・スカラー方式のMPUで行われ、実行ユニットが待ち状態になると、プログラムが指示
15 する命令実行順序に制限されず、実行可能な命令から先に実行していく。しかし、MPU側で勝手に命令実行順序を変更しているので、先に実行した命令が全て無駄になることも多い。

発明の開示

20 本発明の目的は、キャッシュ・メモリのヒット・ミスによるMPUの利用率の低下を低減させることにある。

本発明のマイクロプロセッサは、実行ユニットが実行する命令に、所要のルーチン又はデータ構造体が前記キャッシュ・メモリ内に存在するか否かを確認するテスト命令が含まれる。このようなマイクロプロセッサは、ルーチン又はデータ構造体を読み出す直前にテスト命令を実行す
25 ることにより、読み出そうとしているルーチン又はデータ構造体がキャ

ッシュ・メモリ内に存在するか否かを事前に知ることができる。

- 本発明の命令実行順序スケジューリング方法は、並列処理が可能なルーチン又はデータ構造体を読み出す直前に、そのルーチン又はデータ構造体がキャッシュ・メモリ内に存在するか否かを確認する確認ステップ
- 5 と、並列処理が可能なルーチン又はデータ構造体の中で、キャッシュ・メモリ内に存在することが確認されたルーチン又はデータ構造体から優先的にマイクロプロセッサに処理させる優先実行ステップとを含む。

図面の簡単な説明

- 10 図 1 は、本発明に係るMPUの一構成例を示すブロック図である。
- 図 2 は、本発明に係るスケジューリングの一実施例を示すフロー・チャートである。
- 図 3 は、本発明に係るスケジューリングの他の実施例を示すフロー・チャートである。
- 15 図 4 は、従来のMPUの一構成例を示すブロック図である。
- 図 5 (a) は、MPUが行うデータ処理の一例を示すフロー・チャートであり、図 5 (b)は図 5 (a)中の同一データ構造体を使用するルーチンを続けて実行するフロー・チャート図である。
- 図 6 は、MPUのルーチン実行の一例を示すフロー・チャートである。

20

発明を実施するための最良の態様

次に、本発明に係るマイクロプロセッサ及び命令実行順序スケジューリング方法の実施の形態について、図面に基づいて詳しく説明する。

- 25 図 1 に示すように、本発明に係るMPU 10 の実行ユニット 26 には、指定されたルーチン又はデータ構造体 that キャッシュ・メモリ 14 内に存

在するか確認するテスト命令が追加されている。テスト命令がMPU10に送られると、MPU10はテスト命令が指定するルーチン又はデータ構造体が命令キャッシュ20又はデータ・キャッシュ30に存在するか調べ、存在している(“1”)か存在していない(“0”)かの結果を返す。この結果は汎用レジスタ32に記憶される。テスト命令は、他の命令と同様にMPU10の実行ユニット26で実行される。

プログラムで指示される処理順序をハードウェアの視点に基づいて入れ換えることはできない。しかし、ソフトウェアの視点に基づいて処理順序を入れ換えても問題無く処理を行える場合もある。例えば、依存関係の無い複数のデータ構造体に対して同じ処理を繰り返す場合や、独立の処理を複数行う場合である。本実施形態では、従来(図5(b))のDATA sA, DATA sBを使用したFuncA, FuncBの実行順序のスケジューリングを例にして説明を行う。DATA sA, DATA sBは各々独立したデータ構造体であり、FuncA, FuncBは各々独立したルーチンであり、DATA sA, DATA sBの読み出し、FuncA, FuncBの読み出し及び実行の順序はそれぞれ入れ替え可能である。DATA sA, DATA sB及びFuncA, FuncBはそれぞれプログラムから見たデータ単位及び命令単位である。

図2に、従来(図5(b))のフロー・チャートに本発明のスケジューリング部分(S110, S112, S114, S116, S118, S122, S124, S126, S128, S130)を追加したフロー・チャート図を示す。DATA sA及びDATA sBの読み出し及び処理部分(S172, S174, S182, S176, S178, S186)は従来(図5(b))と同様である。

本発明のスケジューリング方法は、DATA sAを読み出す(S172)直前に、DATA sAがデータ・キャッシュ30に存在するか確認する(S114)。この確認は、テスト命令を用いて行う。テスト命令をMPU10に送

ると、MPU10はDATAsAがデータ・キャッシュ30に存在する(“1”)か存在しない(“0”)かを調べ、その結果(“1” or “0”)を汎用レジスタ32に記憶する。

DATAsAがデータ・キャッシュ30内に存在していれば、DATAsAをデータ・キャッシュ30から読み出して(S172)、DATAsAを使用したFuncA及びFuncBをMPU10に実行させる(S174, S182)。存在していなければ、従来から用いられているタッチ命令をMPU10に送り、DATAsAのプリフェッチを行う(S118)。このプリフェッチは、実行ユニット26の他の命令実行と並行して行うことが可能である。

DATAsBについても、DATAsAと同様に、DATAsBを読み出す(S176)直前に、DATAsBがデータ・キャッシュ30内に存在するかテスト命令で確認する(S124)。DATAsAの場合と同様に、DATAsBがデータ・キャッシュ30内に存在する場合はDATAsBを読み出して(S176)、MPU10にFuncA及びFuncBを実行させる(S178, S186)。存在しない場合はDATAsBをプリフェッチする(S128)。

本発明では、DATAsAを使用したFuncA及びFuncBが完了したか否かを示すパラメータDoneDAと、DATAsBを使用したFuncA及びFuncBが完了したか否かを示すパラメータDoneDBを使用している。DoneDA, DoneDBが“1”の場合はDATAsA, DATAsBを使用したFuncA及びFuncBがそれぞれ完了していることを示し、DoneDA, DoneDBが“0”の場合はDATAsA, DATAsBを使用するFuncA及びFuncBがそれぞれ未完了であることを示す。DoneDA及びDoneDBはデータ・キャッシュ30又は外部メモリ40に記憶される。

DoneDA, DoneDBの初期値は、“0”である(S110)。DATAsAを使用したFuncA及びFuncBが完了すれば、DoneDAは“1”に更新され(S116)、DATAsBを使用したFuncA及びFuncBが完了すれば、DoneDBも“1

”に更新される(S 1 2 6)。DoneDA, DoneDBを参照することにより、DATAs A, DATAs Bを使用したFunc A及びFunc Bが完了したか否かをそれぞれ確認することができる(S 1 1 2, S 1 2 2, S 1 3 0)。

DoneDAを参照したときに(S 1 1 2)DATAs Aを使用したFunc A及びFunc Bが未完了の場合は、DATAs Aがデータ・キャッシュ 3 0内に存在するか確認を行う(S 1 1 4)。同様に、DoneDBを参照したときに(S 1 2 2)DATAs Bを使用したFunc A及びFunc Bが未完了の場合は、DATAs Bがデータ・キャッシュ 3 0内に存在するか確認を行う(S 1 2 4)。

データ又は命令のプリフェッチ中でも、実行ユニット 2 6は他の命令を実行できる。例えば、DoneDAが“0”かつDATAs Aがデータ・キャッシュ 3 0に無く、DoneDBが“0”かつDATAs Bがデータ・キャッシュ 3 0内に存在する場合、DATAs Aのプリフェッチ中にDATAs Bを使用したFunc A及びFunc Bを実行することができる。

次に、DATAs A, DATAs Bを使用したFunc A, Func Bの実行順序のスケジューリングについて、その作用を説明する。

最初にDoneDA及びDoneDBが初期化される(S 1 1 0)。次に、DATAs Aを使用したFunc A及びFunc Bが完了しているか否かを、DoneDAを参照して調べる(S 1 1 2)。DoneDAが“0”の場合は、DATAs Aを使用したFunc A及びFunc Bがまだ実行されていないので、DATAs Aがデータ・キャッシュ 3 0内に存在するか否かをテスト命令で確認する(S 1 1 4)。

DATAs Aがデータ・キャッシュ 3 0内に存在するか否かの判断は、DATAs Aに含まれる全データがデータ・キャッシュ 3 0に存在するか否かで判断することもできるが、簡略化して、DATAs Aの先頭データDATA-A0がデータ・キャッシュ 3 0内に存在するか否かで判断することもできる。DATA-A0がデータ・キャッシュ 3 0に存在すれば、他の部分(DATA-A1, DATA-A2, ……)もデータ・キャッシュ 3 0に存在すると見なすことで、キャ

ッシュ・ヒットとヒット・ミスの判断を簡単かつ高速に行うことができる。

DATA_sAがデータ・キャッシュ30内に存在した場合は、DATA_sAをデータ・キャッシュ30から読み出して(S172)、FuncA、FuncBを実行する(S174, S182)。DATA_sAを使用したFuncA及びFuncBが完了すると、DoneDAを“1”に更新する(S116)。DoneDAが“1”であると、これらDATA_sAに関係する部分(S114, S172, S174, S182, S116, S118)は実行されなくなる(S112)。

DATA_sAがデータ・キャッシュ30内に存在しない場合は、従来のプリフェッチでも使用していたタッチ命令をMPU10に送り(S118)、DATA_sAをプリフェッチする。このDATA_sAのプリフェッチ中に、DATA_sBを使用したFuncA及びFuncBを並行して実行することも可能である。

次のDATA_sBに関係する部分(S122, S124, S176, S178, S186, S126, S128)は、上述したDATA_sAに関係する部分(S112, S114, S172, S174, S182, S116, S118)と同様である。DATA_sBを使用したFuncA及びFuncBが未完了であれば(S122)、DATA_sBがデータ・キャッシュ30内に存在するか否かをテスト命令で確認する(S124)。データ・キャッシュ30内に存在した場合は、DATA_sBをデータ・キャッシュ30から読み出して(S176)、FuncA、FuncBを実行する(S178, S186)。存在しない場合は、DATA_sBのプリフェッチを行う(S128)。

DoneDA及びDoneDBが共に“1”であれば、DATA_sA、DATA_sBを使用したFuncA、FuncBは全て完了している(S130)。従来(図5(b))の場合と異なり、例えばDATA_sAがデータ・キャッシュ30に無く、DATA_sBがデータ・キャッシュ30に存在する場合、DATA_sAのプリフェッチ(S118)と並行して、DATA_sBを使用したFuncA及びFuncBが実行できる(S1

7 8, S 1 8 6)。DATAs Bを使用したFunc A及びFunc Bが完了すると(S 1 2 6)、その処理中にデータ・キャッシュ 3 0にプリフェッチされたDATAs Aを使用したFunc A及びFunc Bを実行する(S 1 7 4, S 1 8 2)。

データ・キャッシュ 3 0に存在しないDATAs Aよりもデータ・キャッシュ 3 0に存在するDATAs Bを先に処理すると共に、DATAs Bの処理と並行してDATAs Aのプリフェッチを行えるため、データ・キャッシュ 3 0のヒット・ミス時のMPU 1 0の待ち時間を短くできる。データ・キャッシュ 3 0のヒット・ミスを確認した後にプリフェッチを行うので、プログラムの実行前の予想に基づいた従来のプリフェッチとは異なり、無駄なプリフェッチは実行されない。

以上、2つのデータ構造体(DATAs A, DATAs B)を例にして説明したが、この並列処理が可能なデータ構造体数は任意である。例えば並列処理が可能なデータ構造体数が5個の場合は、各データ構造体を使用したルーチンが完了しているか否かを示すパラメータを5つ(例えばDoneDA, DoneDB, DoneDC, DoneDD, DoneDE)に増やして、図 2と同様に各データ構造体を読み出す直前にテスト命令を実行し、データ・キャッシュ 3 0内に存在することが確認されたデータ構造体から処理を実行することができる。並列処理が可能なデータ構造体のグループが複数存在する場合もある。データ構造体は、データを1つだけしか含まない場合もある。

以上、データ構造体(DATAs A, DATAs B)を例にして説明したが、ルーチン(Func A, Func B)を実行する際のFunc A, Func Bの読み出しについても、本発明のスケジューリング方法を用いることができる。図 3に、従来(図 6)のフロー・チャートに本発明のスケジューリング部分(S 1 4 0, S 1 4 2, S 1 4 4, S 1 4 6, S 1 4 8, S 1 5 2, S 1 5 4, S 1 5 6, S 1 5 8, S 1 6 0)を追加したフロー・チャートを示す。Func A及びFunc Bの読み出し及び実行部分(S 1 9 0, S 1 9 2, S 1 9 4, S 1 9 6)は従来(

図6)と同様である。

図3において、DoneFAはFuncAが実行されたか否かを示すパラメータであり、DoneFBはFuncBが実行されたか否かを示すパラメータである。DoneFA, DoneFBが“1”の場合はFuncA, FuncBがそれぞれ実行済みであることを示し、DoneFA, DoneFBが“0”の場合はFuncA, FuncBがそれぞれ
5 実行されていないことを示す。

ルーチン(FuncA, FuncB)のスケジューリングは、上述したデータ構造体(DATAsA, DATAsB)のスケジューリングと同様である。FuncA, FuncBの読み出し(S190, S194)前に、FuncA, FuncBが命令キャッシュ
10 20に存在するか否かをテスト命令で確認する(S144, S154)。存在すれば、その命令を命令キャッシュ20から読み出して(S190, S194)、実行する(S192, S196)。存在しなければ、その命令をプリフェッチする(S148, S158)。

FuncAが命令キャッシュ20内に存在するか否かの判断は、FuncAに含まれる全命令が命令キャッシュ20に存在するか否かで判断することもできるが、簡略化して、FuncAの先頭命令Inst-A0が命令キャッシュ20内に存在するか否かで判断することもできる。Inst-A0が命令キャッシュ20に存在すれば、他の部分(Inst-A1, Inst-A2, ……)も命令キャッシュ20に存在すると見なすことで、キャッシュ・ヒットとヒット・ミスの判断を簡単かつ高速に行うことができる。
15 20

従来(図6)の場合と異なり、FuncAが命令キャッシュ20に無く、FuncBが命令キャッシュ20に存在する場合は、FuncAのプリフェッチ(S148)と並行して、FuncBを実行することができる(S196)。FuncBが実行し終わると(S156)、FuncBが実行されている間に命令キャッシュ20にプリフェッチされたFuncAを実行する(S192)。
25

命令キャッシュ20に存在しないFuncAよりも命令キャッシュ20に

存在するFunc Bを先に実行すると共に、Func Bの実行と並行してFunc Aのプリフェッチを行えるため、命令キャッシュ20のヒット・ミス時のMPU10の待ち時間を短くできる。命令キャッシュ20のヒット・ミスを確認した後にプリフェッチを行うので、プログラム実行前の予想に基づいた従来のプリフェッチとは異なり、無駄なプリフェッチは実行されない。

以上、2つのルーチン(Func A, Func B)を例にして説明したが、この並列実行が可能なルーチン数は任意である。例えば並列実行が可能なルーチン数が5個の場合は、各ルーチンの実行が終了しているか否かを示すパラメータを5つ(例えばDoneFA, DoneFB, DoneFC, DoneFD, DoneFE)に増やして、図3と同様に各ルーチンの読み出し前にテスト命令を実行し、命令キャッシュ20内に存在することが確認されたルーチンから実行していくことができる。並列処理が可能なルーチンのグループが複数存在する場合もある。ルーチンは、命令を1つだけしか含まない場合もある。

上述したデータ構造体の読み出し時のスケジューリングとルーチンの読み出し時のスケジューリングとは、任意に組み合わせることができる。例えば、図2に示したFunc A及びFunc Bの読み出しに、図3に示したスケジューリングを用いることもできる。命令キャッシュ20とデータ・キャッシュ30とは各々独立しているので、命令キャッシュ20の読み書きとデータ・キャッシュ30の読み書きとはそれぞれ独立して実行できる。

図5(b)及び図6に示す従来のフロー・チャートに追加された本発明のスケジューリング部分は、従来のルーチン又はデータ構造体の読み出し及び処理に関係しない部分には一切分岐していない。本発明で追加したスケジューリング部分がフロー・チャートの他の部分に影響を与えることはない。本発明は、データ構造体及びルーチンの順序入れ換えをブ

プログラムで指示している。OSのスケジューラで入れ替えを制御するマルチ・スレッドと異なり、コンテキスト・スイッチを起動させないため、MPU及びOSの負担も少なく高速処理が可能である。

- 並列処理が可能なルーチン又はデータ構造体を読み出す直前部分への
- 5 テスト命令等の追加は、コンパイル時に自動で追加できる場合と、ソース・プログラムに手動で追加する場合がある。一般に、行列演算等のアルゴリズムの場合はコンパイラが並行性を検出できるので、コンパイル時にテスト命令等を自動的に追加できることが多い。I/O(入力/出力)に関係する部分は、コンパイラで並行性を検出できないので、手動でテスト命令等を追加することが多い。
- 10

- 以上、本発明は特定の実施例について説明されたが、本発明はこれらに限定されるものではない。例えば、アドレス変換時に使用されるTLB(Translation Lookaside Buffer)にテスト命令を送ることもできる。TLBは、外部メモリ40に存在するアドレス変換テーブルの一部が記憶されるキャッシュ・メモリである。TLBに格納されたアドレス変換
- 15 テーブルの入れ替えは、PTE(Page Table Entry)の自動検索によって行われる。TLBを使用する場合、キャッシュ・ヒットは、命令キャッシュ(又はデータ・キャッシュ)とTLBの両方がキャッシュ・ヒットしたことを意味する。

- 20 PTEの自動検索は、外部メモリ40を複数回アクセスしながらTLBの格納データの入れ替えを行う。本発明のMPU及び命令実行順序スケジューリング方法を用いることにより、データ・キャッシュ及び命令キャッシュのヒット・ミスの場合と同様に、TLBのヒット・ミスの場合もMPUの待ち時間を短くすることができる。テスト命令は、命令キャッシュ、データ・キャッシュ、TLBの各キャッシュ・ヒットを一括して調べてもよいし、各々のキャッシュ・ヒットを独立に調べてもよい。
- 25

その他、本発明はその趣旨を逸脱しない範囲で当業者の知識に基づき種々なる改良、修正、変形を加えた態様で実施できるものである。

本発明のマイクロプロセッサは、所要のルーチン又はデータ構造体がキャッシュ・メモリ(命令キャッシュ, データ・キャッシュ)に存在するか
5 確認する命令(テスト命令)が追加されている。テスト命令により、これから読み出すルーチン又はデータ構造体がキャッシュ・メモリに存在するか否かを、そのルーチン又はデータ構造体を読み出す直前に知ることができる。

本発明のスケジューリング方法は、上述したテスト命令により、並列
10 処理が可能なルーチン又はデータ構造体に対して、これから読み出すルーチン又はデータ構造体がキャッシュ・メモリに存在するか否かを調べ、その結果に基づいて、キャッシュ・メモリに存在するルーチン又はデータ構造体を優先的に読み出す。キャッシュ・メモリに存在しないルーチン又はデータ構造体よりもキャッシュ・メモリに存在するルーチン又は
15 データ構造体を先に読み出して処理すると共に、その処理と並行してキャッシュ・メモリに存在しないルーチン又はデータ構造体のプリフェッチを行うため、キャッシュ・メモリのヒット・ミスにより生じるマイクロプロセッサの待ち時間を短くすることができる。

請求の範囲

1. 外部メモリから読み出される命令の一部及びデータの一部が記憶さ
5 れるキャッシュ・メモリと、キャッシュ・メモリ又は外部メモリから読
み出された命令又は読み出されたデータを使用する命令を実行する実行
ユニットとを備え、プログラムに指示された順序で命令を処理するマイ
クロプロセッサであって、

前記実行ユニットが実行する命令に、複数の命令を含む所要のルーチ
10 ン又は複数のデータを含む所要のデータ構造体が前記キャッシュ・メモ
リ内に存在するか否かを確認するテスト命令が含まれるマイクロプロセ
ッサ。

2. 前記テスト命令が、前記所要のルーチン又はデータ構造体の先頭ア
ドレス部分が前記キャッシュ・メモリ内に存在するか否かを確認する命
15 令を含む請求項1のマイクロプロセッサ。

3. 前記キャッシュ・メモリが、
ルーチンが記憶される命令キャッシュ・メモリと、
データ構造体が記憶されるデータ・キャッシュ・メモリと
を含む請求項1又は請求項2のマイクロプロセッサ。

20 4. 前記テスト命令が、前記所要のルーチンが前記命令キャッシュ・メ
モリ内に存在するか否かを確認する命令と、

前記所要のデータ構造体が前記データ・キャッシュ・メモリ内に存在
するか否かを確認する命令と
を含む請求項3のマイクロプロセッサ。

25 5. 前記実行ユニットの命令実行と並行して、所要のルーチン又はデー
タ構造体の読み出しを行うフェッチ・ユニットをさらに含む請求項1乃

至請求項 4 のいずれかのマイクロプロセッサ。

6. プログラムが指定する順序で、外部メモリ又はキャッシュ・メモリから読み出された複数の命令を含むルーチン又は複数のデータを含むデータ構造体を使用するルーチンをマイクロプロセッサに実行させる場合
- 5 の、前記ルーチン又はデータ構造体に含まれる並列処理が可能なルーチン又はデータ構造体に対する命令実行順序のスケジューリング方法であって、

- 並列処理が可能なルーチン又はデータ構造体を読み出す直前に、そのルーチン又はデータ構造体がキャッシュ・メモリ内に存在するか否かを
- 10 確認する確認ステップと、

前記並列処理が可能なルーチン又はデータ構造体の中で、キャッシュ・メモリ内に存在することが確認されたルーチン又はデータ構造体から優先的にマイクロプロセッサに処理させる優先実行ステップとを含む命令実行順序スケジューリング方法。

7. 前記優先実行ステップが、読み出そうとしているルーチン又はデータ構造体がキャッシュ・メモリ内に存在すれば、そのルーチン又はデータ構造体をキャッシュ・メモリから読み出してマイクロプロセッサに処理させる実行ステップと、
- 15

- 読み出そうとしているルーチン又はデータ構造体がキャッシュ・メモリ内に存在しなければ、そのルーチン又はデータ構造体を外部メモリから読み出すようマイクロプロセッサに指示する指示ステップと
- 20
- を含む請求項 6 の命令実行順序スケジューリング方法。

8. 前記指示ステップのルーチン又はデータ構造体の外部メモリからの読み出しが、前記実行ステップと並行して行われる請求項 7 の命令実行
- 25
- 順序スケジューリング方法。

9. 前記確認ステップが、前記実行ステップ又は指示ステップの後、処

理が未完了のルーチン又はデータ構造体が存在するか確認するステップと、

- 処理が未完了のルーチン又はデータ構造体が存在すれば、そのルーチン又はデータ構造体がキャッシュ・メモリ内に存在するかマイクロプロセッサに確認させるステップと
- 5

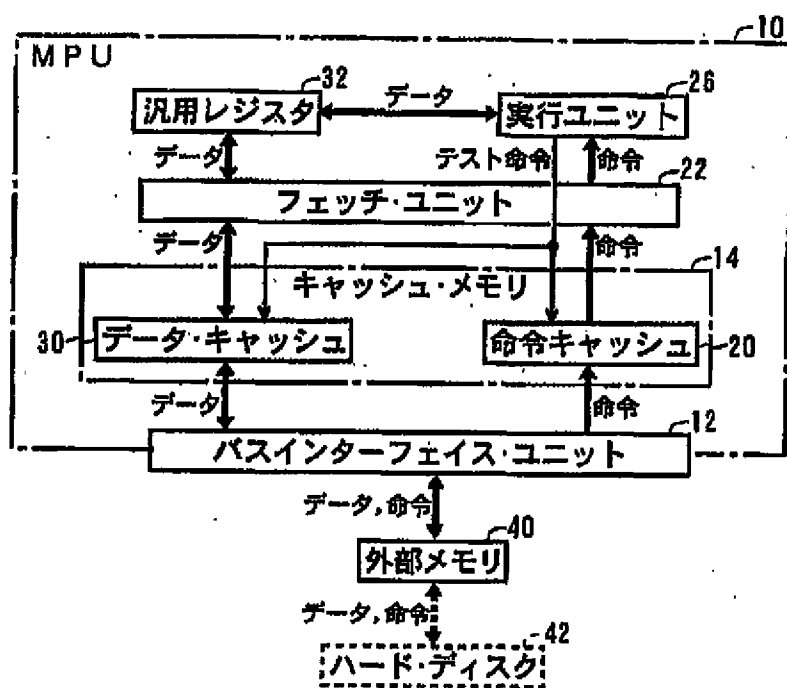
を含む請求項 6 乃至請求項 8 のいずれかの命令実行順序スケジューリング方法。

- 10 10. 前記実行ステップが、処理が完了したルーチン又はデータ構造体に対し、ルーチン又はデータ構造体が処理されたか否かを示す実行完了情報を更新するステップを含み、

前記処理が未完了のルーチン又はデータ構造体が存在するか確認するステップが、前記実行完了情報に基づいて処理が未完了のルーチン又はデータ構造体が存在するか確認を行う請求項 7 乃至請求項 9 のいずれかの命令実行順序スケジューリング方法。

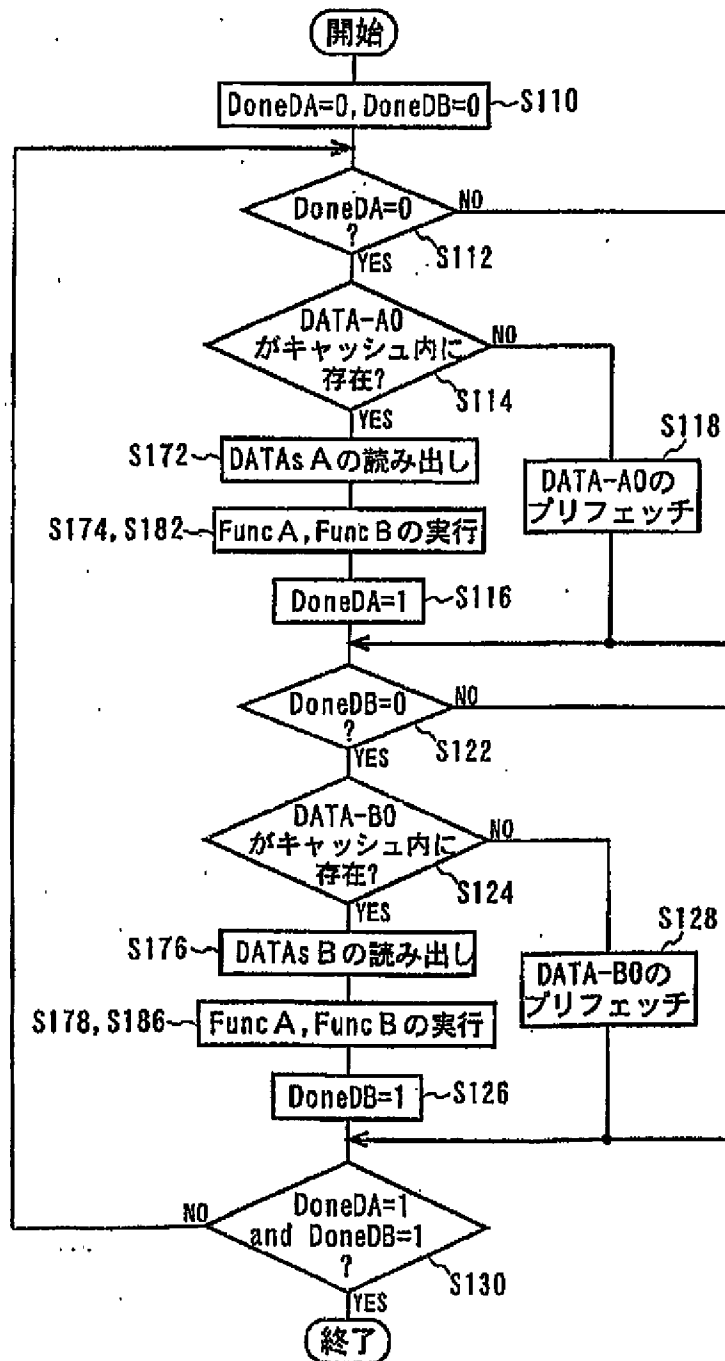
- 15 11. 前記確認ステップが、前記ルーチン又はデータ構造体の先頭アドレス部分に基づいてキャッシュ・メモリ内の存在確認を行う請求項 6 乃至請求項 9 のいずれかの命令実行順序スケジューリング方法。

1/6



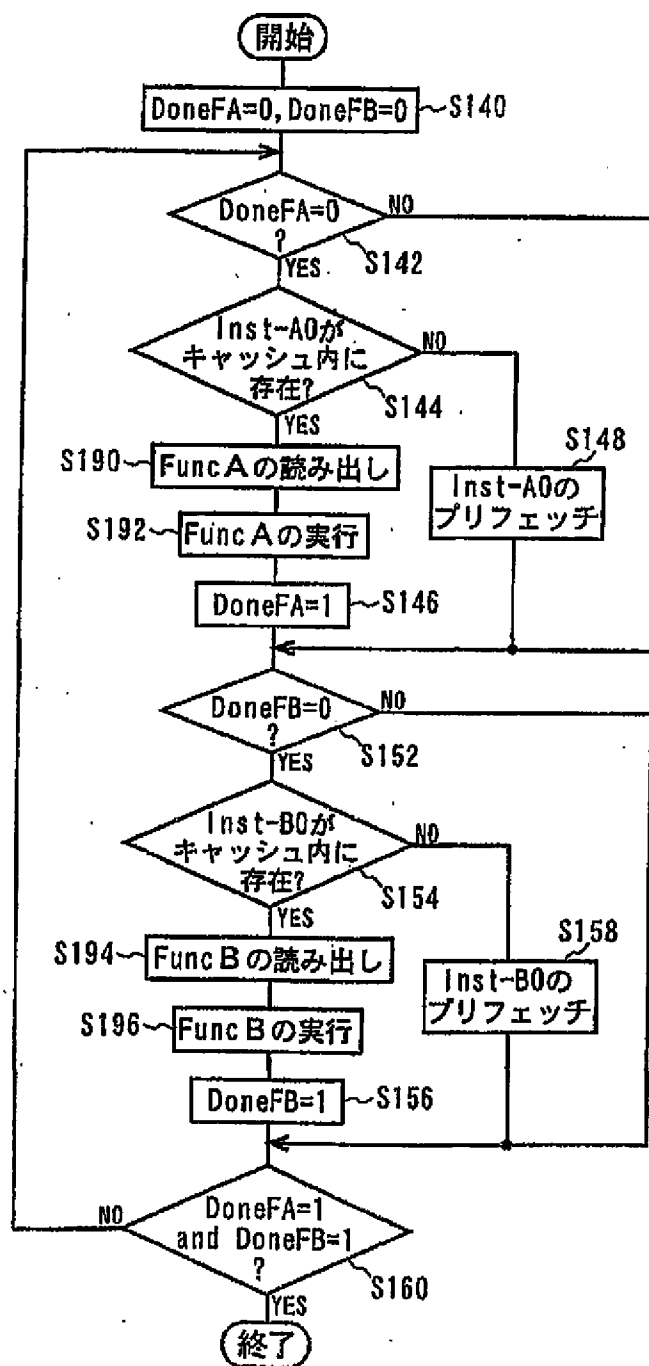
第 1 図

2/6



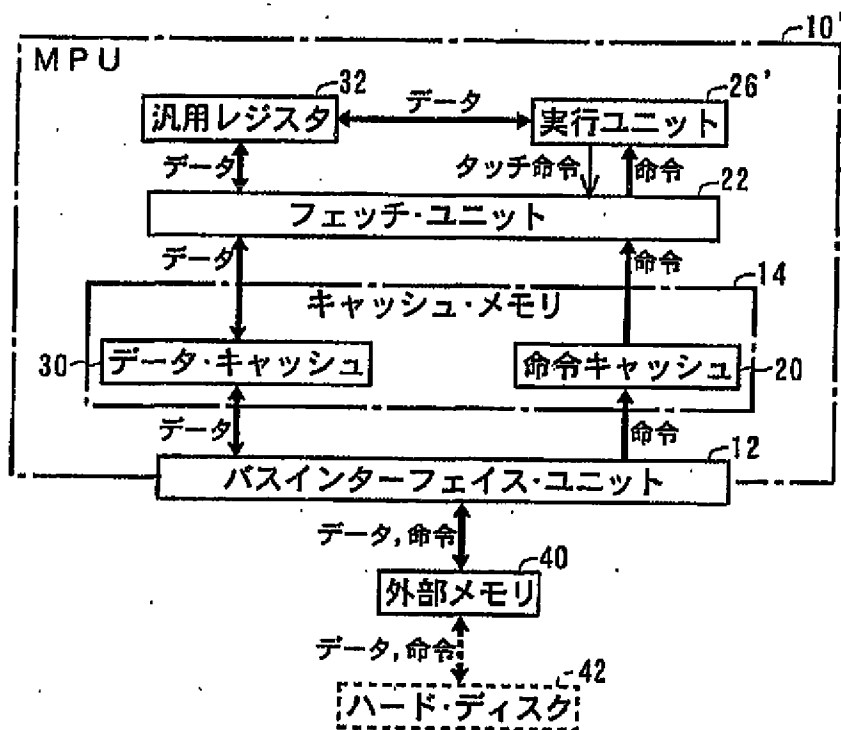
第 2 図

3/6



第 3 図

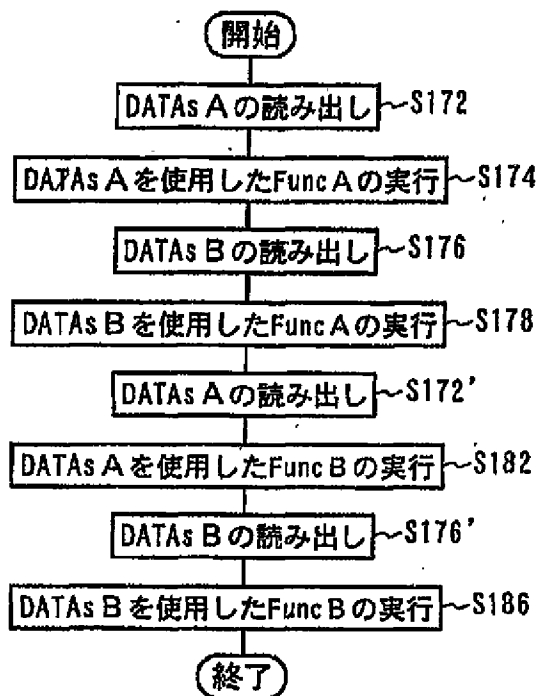
4/6



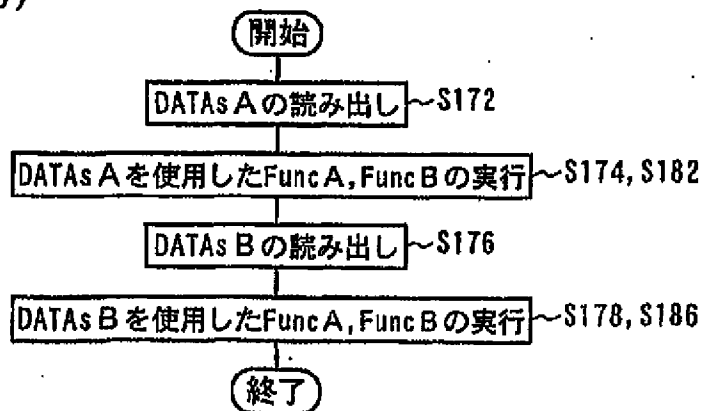
第 4 図

5/6

(a)

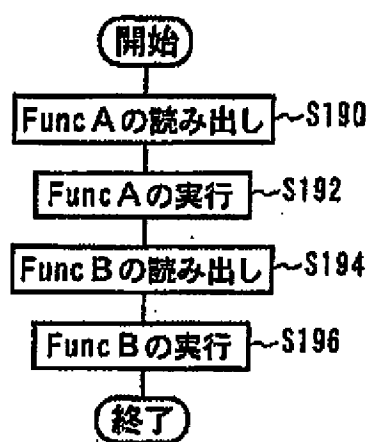


(b)



第 5 図

6/6



第 6 図

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP02/01272

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl⁷ G06F12/08, G06F9/30, G06F9/38, G06F9/32, G06F9/34

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁷ G06F12/08, G06F9/30, G06F9/38, G06F9/32, G06F9/34

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Jitsuyo Shinan Toroku Koho	1996-2002
Kokai Jitsuyo Shinan Koho	1971-2002	Toroku Jitsuyo Shinan Koho	1994-2002

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	HOROWITZ et al., "Informing Memory Operation: Memory Performance Feedback Mechanizms and Their Applications" ACM Trans. On Computer Systems, ACM, 1998, Vol.16, No.2, pages 170 to 205 Paragraph 2, 4, 5	1-11

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
04 March, 2002 (04.03.02)

Date of mailing of the international search report
12 March, 2002 (12.03.02)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP02/01272

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

The feature common to all claims 1-11 is the technique of "including a test instruction to check if a desired routine including instructions or a desired data structure containing data is present in cache memories in the instructions executed by an execution unit". However the search has revealed that this feature is not novel since it is disclosed in the document below. Consequently the common feature is not a special technical feature within the meaning of PCT Rule 13.2, second sentence. Therefore claims 1-11 do not satisfy the requirement of unity of invention under PCT Rule 13.

HOROWITZ et al. 'Informing Memory Operation: Memory Performance Feedback Mechanisms and Their Applications' ACM Trans. on Computer Systems, ACM, 1998, Vol. 16, No. 2, pp. 170-205

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☒ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest ☐ The additional search fees were accompanied by the applicant's protest.
☐ No protest accompanied the payment of additional search fees.

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. Cl⁷ G06F12/08, G06F9/30, G06F9/38, G06F9/32, G06F9/34

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. Cl⁷ G06F12/08, G06F9/30, G06F9/38, G06F9/32, G06F9/34

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1922-1996年

日本国公開実用新案公報 1971-2002年

日本国実用新案登録公報 1996-2002年

日本国登録実用新案公報 1994-2002年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X	HOROWITZ et al.' Informing Memory Operation: Memory Performance Feedback Mechanisms and Their Applications' ACM Trans. on Computer Systems, ACM, 1998, Vol. 16, No. 2, p. 170-205 第2、第4、第5パラグラフ	1-11

☐ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの

「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)

「O」 口頭による開示、使用、展示等に言及する文献

「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」 同一パテントファミリー文献

国際調査を完了した日

04.03.02

国際調査報告の発送日

12.03.02

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JP)

郵便番号 100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

相崎 裕恒

5N

9290

電話番号 03-3581-1101 内線 3585

第I欄 請求の範囲の一部の調査ができないときの意見 (第1ページの2の続き)

法第8条第3項 (PCT17条(2)(a)) の規定により、この国際調査報告は次の理由により請求の範囲の一部について作成しなかった。

1. ☐ 請求の範囲 _____ は、この国際調査機関が調査をすることを要しない対象に係るものである。つまり、
2. ☐ 請求の範囲 _____ は、有意義な国際調査をすることができる程度まで所定の要件を満たしていない国際出願の部分に係るものである。つまり、
3. ☐ 請求の範囲 _____ は、従属請求の範囲であってPCT規則6.4(a)の第2文及び第3文の規定に従って記載されていない。

第II欄 発明の単一性が欠如しているときの意見 (第1ページの3の続き)

次に述べるようにこの国際出願に二以上の発明があるとこの国際調査機関は認めた。

請求の範囲1-11のすべてに共通する技術的事項である「実行ユニットが実行する命令に、複数の命令を含む所要のルーチン又は複数のデータを含む所要のデータ構造体がキャッシュメモリ内に存在するか否かを確認するテスト命令が含まれる」点は、下記文献によって開示されているから、新規でなく、PCT規則13.2第2センテンスの意味で特別な技術的特徴でない。したがって、請求の範囲1-11はPCT規則13に規定する発明の単一性の要件を満たしていない。

HOROWITZ et al. 'Informing Memory Operation: Memory Performance Feedback Mechanizms and Their Applications' ACM Trans. on Computer Systems, ACM, 1998, Vol. 16, No. 2, p. 170-205

1. ☐ 出願人が必要な追加調査手数料をすべて期間内に納付したので、この国際調査報告は、すべての調査可能な請求の範囲について作成した。
2. ☒ 追加調査手数料を要求するまでもなく、すべての調査可能な請求の範囲について調査することができたので、追加調査手数料の納付を求めなかった。
3. ☐ 出願人が必要な追加調査手数料を一部のみしか期間内に納付しなかったため、この国際調査報告は、手数料の納付のあった次の請求の範囲のみについて作成した。
4. ☐ 出願人が必要な追加調査手数料を期間内に納付しなかったため、この国際調査報告は、請求の範囲の最初に記載されている発明に係る次の請求の範囲について作成した。

追加調査手数料の異議の申立てに関する注意

- ☐ 追加調査手数料の納付と共に出願人から異議申立てがあった。
☐ 追加調査手数料の納付と共に出願人から異議申立てがなかった。